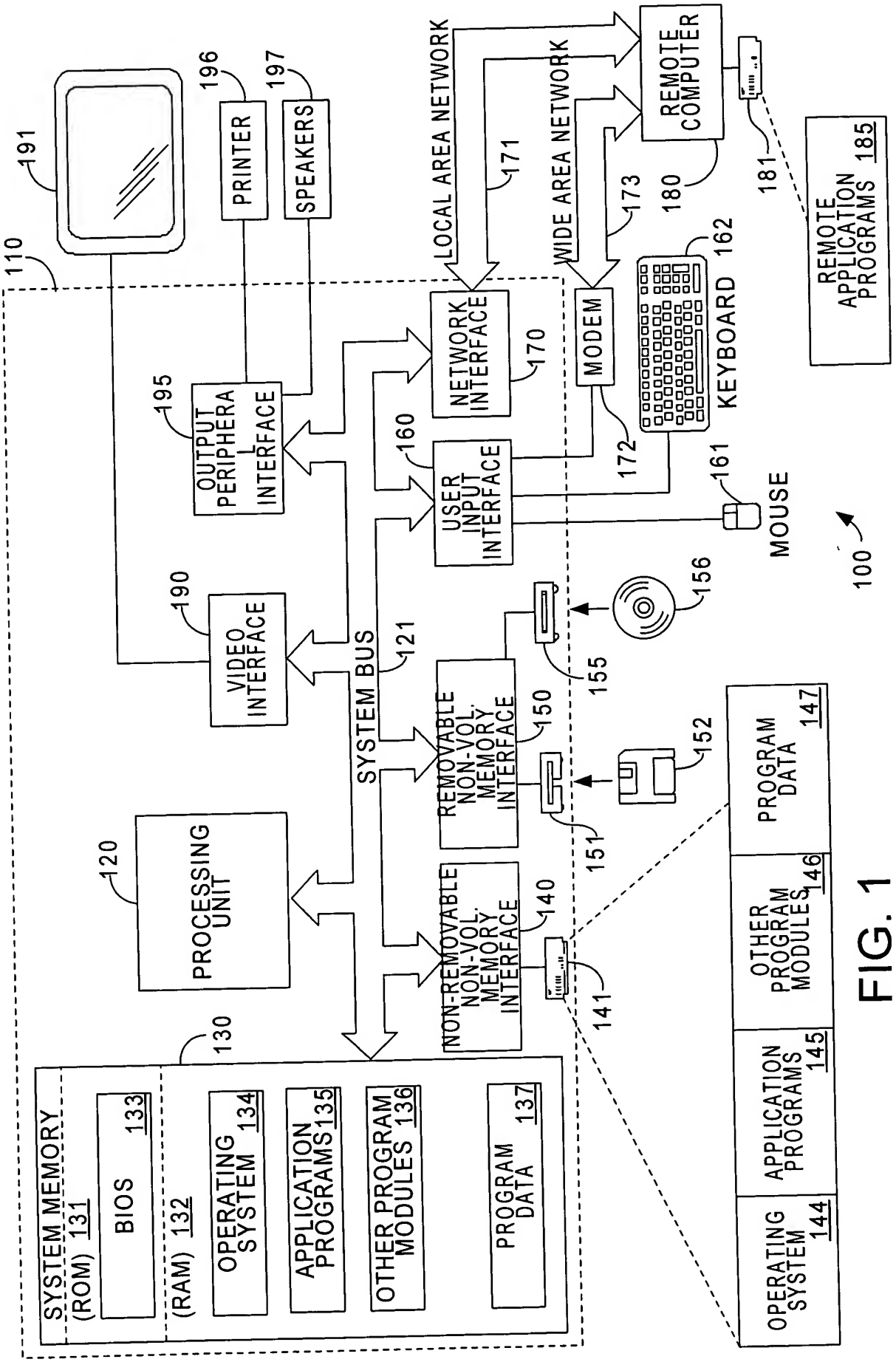

110
191
196
PRINTER
197
SPEAKERS
LOCAL AREA NETWORK
171
WIDE AREA NETWORK
REMOTE COMPUTER
173
180
181
REMOTE APPLICATION PROGRAMS 185
162
KEYBOARD
161
MOUSE
172
MODEM
NETWORK INTERFACE
170
195
OUTPUT PERIPHERAL INTERFACE
160
USER INPUT INTERFACE
190
VIDEO INTERFACE
121
SYSTEM BUS
156
155
150
REMOVABLE NON-VOL. MEMORY INTERFACE
152
151
140
NON-REMOVABLE NON-VOL. MEMORY INTERFACE
141
120
PROCESSING UNIT
130
SYSTEM MEMORY
(ROM) 131
BIOS 133
(RAM) 132
OPERATING SYSTEM 134
APPLICATION PROGRAMS 135
OTHER PROGRAM MODULES 136
PROGRAM DATA 137
OPERATING SYSTEM 144
APPLICATION PROGRAMS 145
OTHER PROGRAM MODULES 146
PROGRAM DATA 147
100

FIG. 1

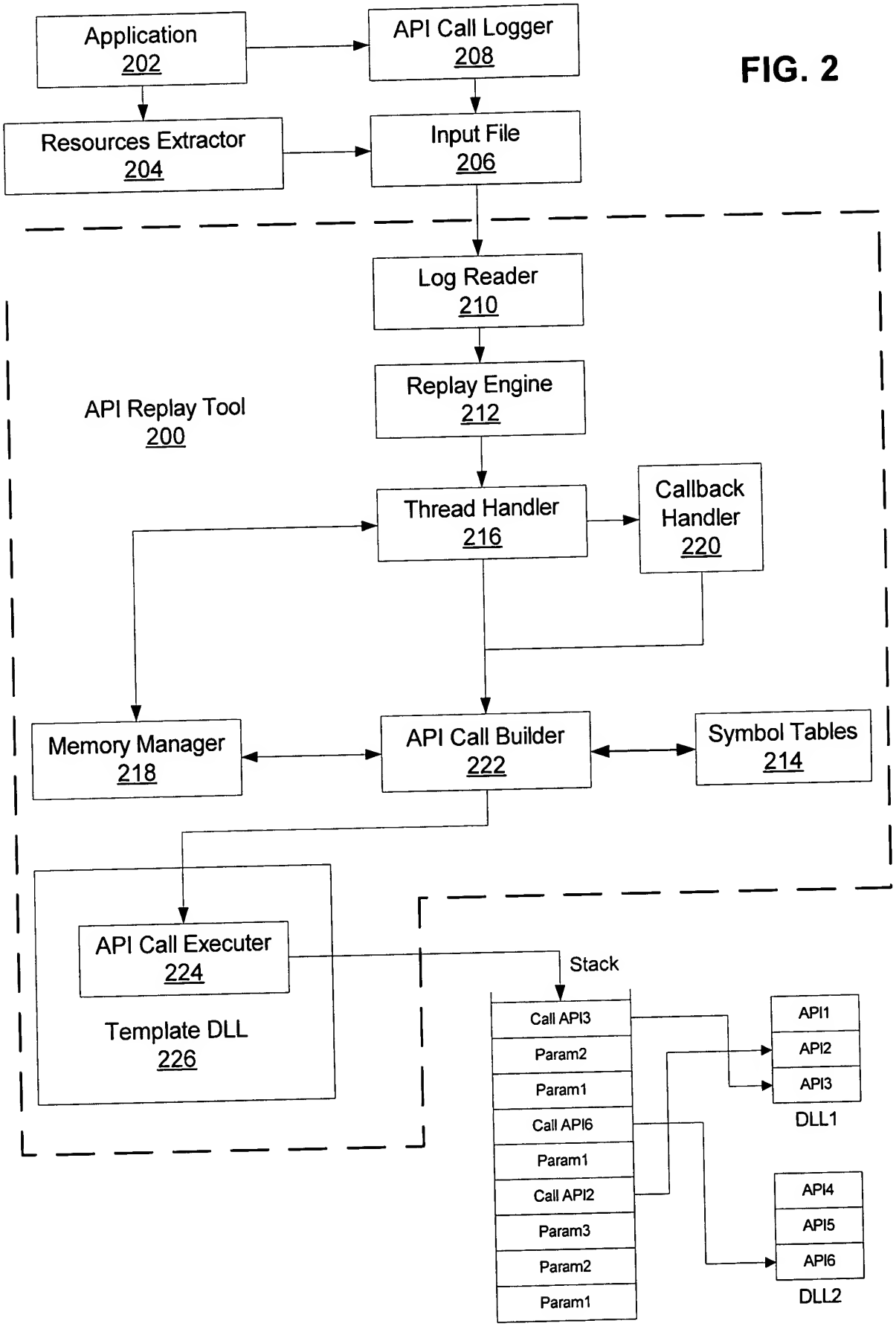

Application
202
API Call Logger
208
Resources Extractor
204
Input File
206
Log Reader
210
Replay Engine
212
API Replay Tool
200
Thread Handler
216
Callback Handler
220
Memory Manager
218
API Call Builder
222
Symbol Tables
214
API Call Executer
224
Template DLL
226
Stack
Call API3
Param2
Param1
Call API6
Param1
Call API2
Param3
Param2
Param1
API1
API2
API3
DLL1
API4
API5
API6
DLL2

FIG. 2

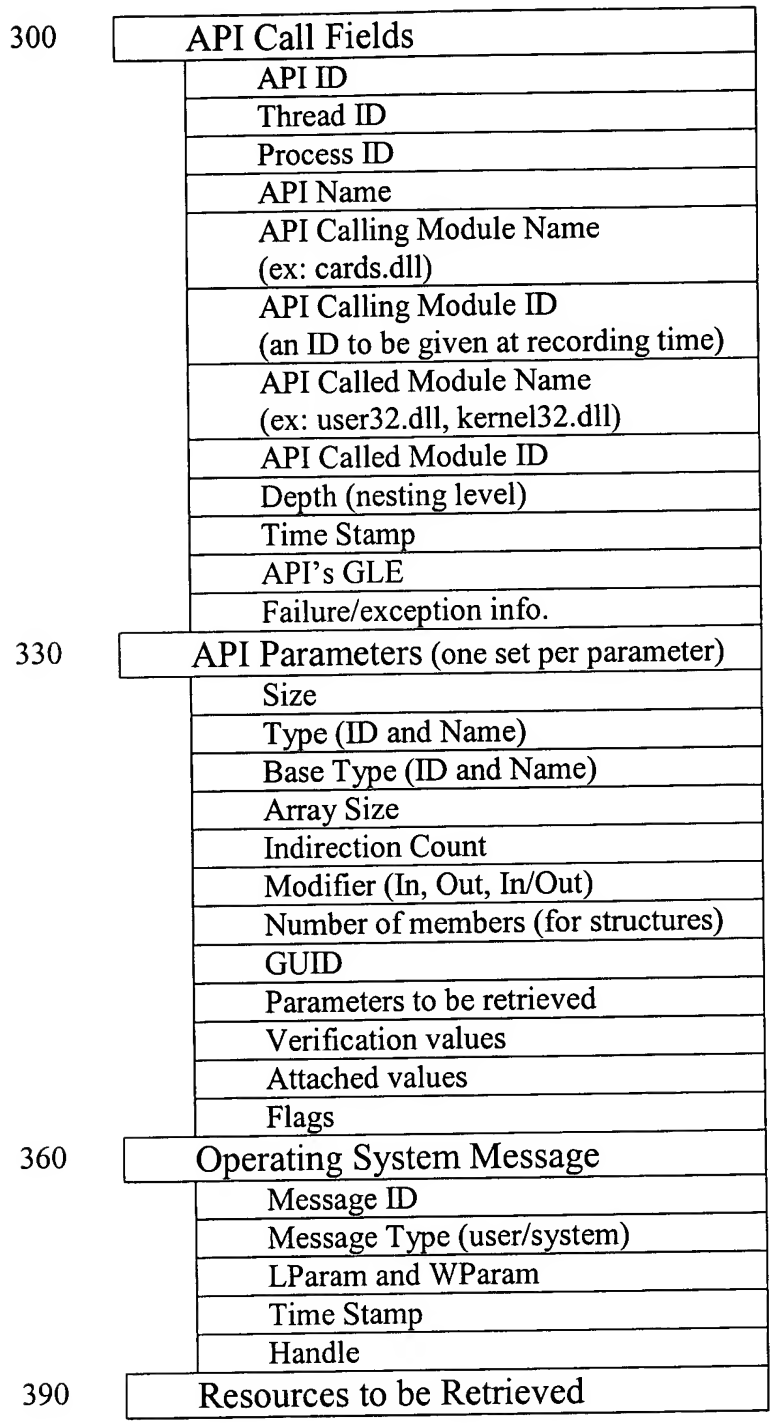

300
API Call Fields
API ID
Thread ID
Process ID
API Name
API Calling Module Name
(ex: cards.dll)
API Calling Module ID
(an ID to be given at recording time)
API Called Module Name
(ex: user32.dll, kernel32.dll)
API Called Module ID
Depth (nesting level)
Time Stamp
API's GLE
Failure/exception info.
330
API Parameters (one set per parameter)
Size
Type (ID and Name)
Base Type (ID and Name)
Array Size
Indirection Count
Modifier (In, Out, In/Out)
Number of members (for structures)
GUID
Parameters to be retrieved
Verification values
Attached values
Flags
360
Operating System Message
Message ID
Message Type (user/system)
LParam and WParam
Time Stamp
Handle
390
Resources to be Retrieved

FIG. 3

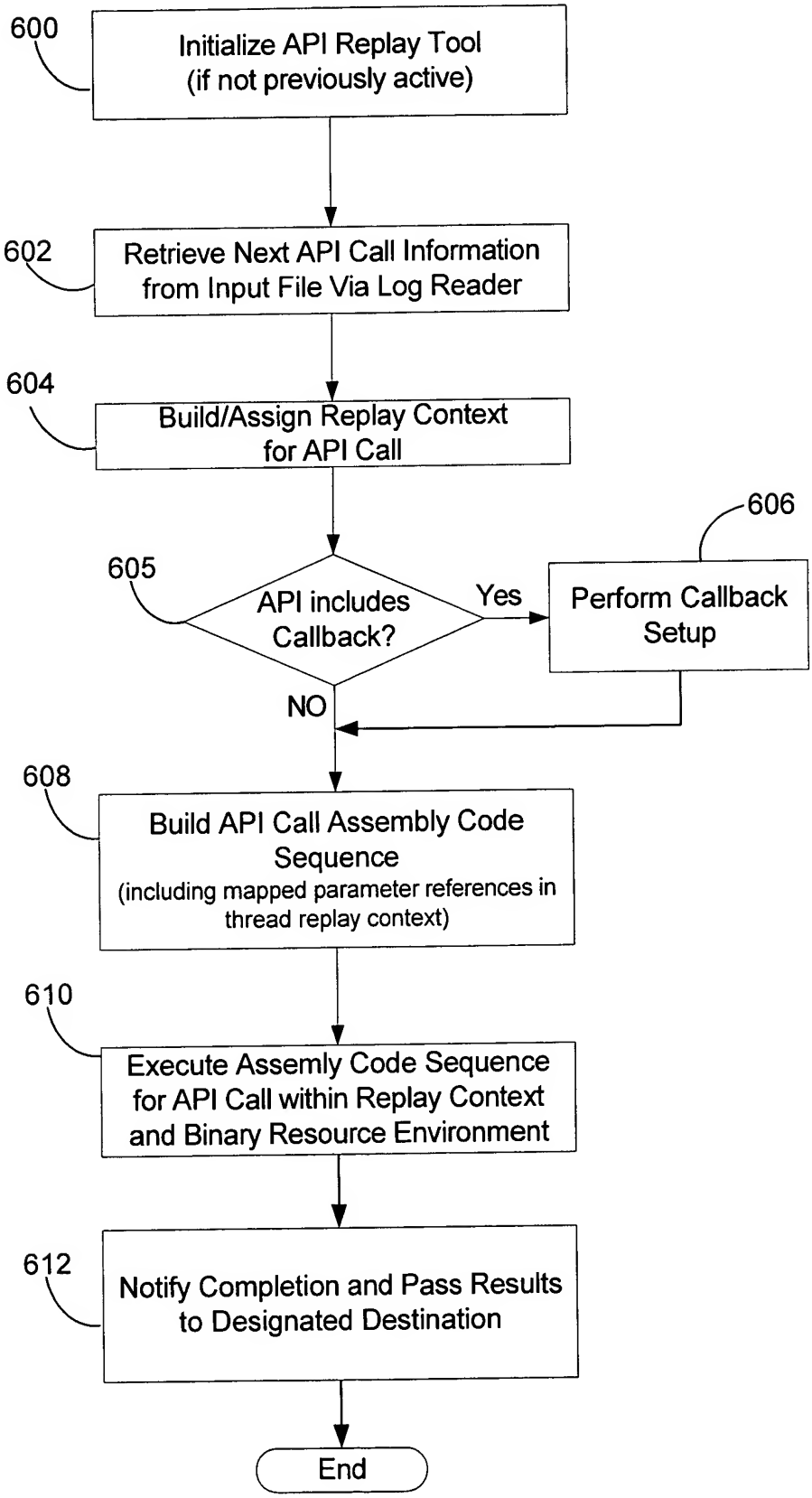
400	Initialize
402	RunLog
404	CreateInitialProcess
406	CreateInitialThread
408	HandleSpecialAPI
410	IsFilteredAPI
412	GetFirstLogRecord
414	GetNextAPICode
416	GetNextLogRecord
418	GetNextBlockCode
420	GetNextSegmentCode
422	GetInterface(*)

FIG. 4

500	Memory Manager
510	Process Block
520	Thread Block
530	Memory Block
540	Code Data

FIG. 5

FIG. 6


600
Initialize API Replay Tool (if not previously active)
602
Retrieve Next API Call Information from Input File Via Log Reader
604
Build/Assign Replay Context for API Call
606
605
API includes Callback?
Yes
Perform Callback Setup
NO
608
Build API Call Assembly Code Sequence
(including mapped parameter references in thread replay context)
610
Execute Assemly Code Sequence for API Call within Replay Context and Binary Resource Environment
612
Notify Completion and Pass Results to Designated Destination
End